Amendments to the Claims:

This listing of claims replaces all prior versions and listings of claims in the application.

Listing of Claims:

1.   (Currently Amended) A method for dynamically programming a field programmable gate array (FPGA) in a coprocessor, the method comprising:

providing a processor and a coprocessor that is separate from the processor in that the coprocessor is coupled to the processor through a bus that is external to the processor, the coprocessor being coupled to the processor and including a field programmable gate array (FPGA);

executing an application using the processor;

the coprocessor receiving an instruction from the processor to perform a function for the application;

determining that the field programmable gate array (FPGA) in the coprocessor is not programmed to perform the function for the application;

fetching a configuration bit stream associated with the function for the application responsive to the determination that the field programmable gate array (FPGA) is not programmed to perform the function for the application; and

dynamically programming the field programmable gate array (FPGA) in accordance with the configuration bit stream to perform the function for the application.

2.   (Previously Presented) The method of claim 1, wherein the coprocessor further comprises an Auxiliary Processing Unit (APU) interface for receiving instructions from the processor, the Auxiliary Processing Unit (APU) interface determining whether a given

instruction is to be processed by the coprocessor.

3.   (Previously Presented) The method of claim 2, wherein the Auxiliary Processing Unit (APU) interface determining whether a given instruction is to be processed by the coprocessor includes the Auxiliary Processing Unit (APU) interface issuing a faulty commit if the given instruction is to be processed by the coprocessor and the field programmable gate array (FPGA) in the coprocessor is not programmed to perform a function corresponding to the given instruction.

4.   (Previously Presented) The method of claim 3, further comprising:

the processor initiating an exception subroutine responsive to the Auxiliary Processing Unit (APU) interface issuing a faulty commit; and

the exception routine identifying the configuration bit stream associated with the function for the application.

5.   (Previously Presented) The method of claim 4, wherein the the processor initiating an exception subroutine comprises the processor branching to the exception subroutine in response to the faulty commit.

6.   (Previously Presented) The method of claim 4, wherein the exception routine identifying the configuration bit stream comprises:

the exception routine decoding a function identifier associated with the function corresponding to the given instruction;

the processor fetching the configuration bit stream ~~for~~ associated with the function corresponding to the given instruction from a memory; and

the processor sending the configuration bit stream associated with the function corresponding to the given instruction to a direct memory access (DMA) channel coupled to the coprocessor for programming the field programmable gate array (FPGA) in the coprocessor.

7.   (Previously Presented) The method of claim 1, wherein dynamically programming the field programmable gate array (FPGA) comprises the processor performing a sequence of load and store instructions in accordance with an exception subroutine of the processor to program the field programmable gate array (FPGA) in accordance with the configuration bit stream.

8.   (Previously Presented) The method of claim 1, further comprising the processor reissuing the instruction to the coprocessor responsive to the field programmable gate array (FPGA) being dynamically reconfigured to perform the function for the application.

9.   (Currently Amended) A method for dynamically programming a field programmable gate array (FPGA) in a coprocessor, the method comprising:

providing a processor and a coprocessor that is separate from the processor <u>in that the coprocessor is coupled to the processor through a bus that is external to the processor</u>, the coprocessor ~~being coupled to the processor and~~ including a field programmable gate array (FPGA) and an Auxiliary Processing Unit (APU) interface;

executing an application using the processor;

the coprocessor receiving an instruction from the processor to perform a function for the

application;

the Auxiliary Processing Unit (APU) interface issuing a faulty commit when the field programmable gate array (FPGA) in the coprocessor is not programmed to perform the function;

the processor initiating an exception subroutine in response to the faulty commit;

the exception subroutine fetching a configuration bit stream associated with the function for the application; and

the exception subroutine performing a sequence of load and store instructions to program the field programmable gate array (FPGA) in accordance with the configuration bit stream to perform the function.

10.   (Currently Amended) A computer readable medium with program instruction tangibly stored thereon for dynamically programming a field programmable gate array (FPGA) in a coprocessor, the coprocessor being coupled to a processor that is separate from the coprocessor in that the coprocessor is coupled to the processor through a bus that is external to the processor, the computer readable medium comprising instructions for:

executing an application using the processor;

the coprocessor receiving an instruction from the processor to perform a function for the application;

determining that the field programmable gate array (FPGA) in the coprocessor is not programmed to perform the function for the application;

fetching a configuration bit stream ~~for~~ associated with the function for the application; and

programming the field programmable gate array (FPGA) in accordance with the

configuration bit stream to perform the function for the application.

11.   (Previously Presented) The computer readable medium of claim 10, wherein the coprocessor further includes an Auxiliary Processing Unit (APU) interface for receiving instructions from the processor, and the computer readable medium further comprises instructions for the Auxiliary Processing Unit (APU) interface determining whether a given instruction is to be processed by the coprocessor.

12.   (Previously Presented) The computer readable medium of claim 11, wherein the instructions for the Auxiliary Processing Unit (APU) interface determining whether a given instruction is to be processed by the coprocessor comprises ~~the~~ instructions for the Auxiliary Processing Unit (APU) interface issuing a faulty commit if the given instruction is to be processed by the coprocessor and the field programmable gate array (FPGA) in the coprocessor is not programmed to perform a function corresponding to the given instruction.

13.   (Previously Presented) The computer readable medium of claim 12, further comprising instructions for:

the processor initiating an exception subroutine responsive to the Auxiliary Processing Unit (APU) interface issuing a faulty commit; and

the exception routine identifying the configuration bit stream associated with the function for the application.

14.   (Previously Presented) The computer readable medium of claim 13, wherein the

instructions for processor initiating an exception subroutine comprises ~~the~~ instructions for the processor branching to the exception subroutine in response to the faulty commit.

15.   (Previously Presented) The computer readable medium of claim 13, wherein the instructions for the exception routine identifying the configuration bit stream comprises instructions for:

the exception routine decoding a function identifier associated with the function corresponding to the given instruction;

the processor fetching the configuration bit stream ~~for~~ associated with the function corresponding to the given instruction from a memory; and

the processor sending the configuration bit stream associated with the function corresponding to the given instruction to a direct memory access (DMA) channel coupled to the coprocessor for programming the field programmable gate array (FPGA) in the coprocessor.

16.   (Previously Presented) The computer readable medium of claim 10, wherein the instructions for dynamically programming the field programmable gate array (FPGA) comprises instructions for the processor performing a sequence of load and store instructions in accordance with an exception subroutine of the processor to program the field programmable gate array (FPGA) in accordance with the configuration bit stream.

17.   (Previously Presented) The computer readable medium of claim 10, further comprising ~~the~~ instructions for the processor reissuing the instruction to the coprocessor responsive to the field programmable gate array (FPGA) being dynamically reconfigured to perform the function

for the application.

18.   (Currently Amended) A computer readable medium with programming instructions tangibly stored thereon for dynamically programming a field programmable gate array (FPGA) in a coprocessor, the coprocessor including an Auxiliary Processing Unit (APU) interface and being coupled to a processor that is separate from the coprocessor <u>in that the coprocessor is coupled to the processor through a bus that is external to the processor</u>, the computer readable medium comprising instructions for:

        executing an application using the processor;

        the coprocessor receiving an instruction from the processor to perform a function for the application;

        the Auxiliary Processing Unit (APU) interface issuing a faulty commit when the field programmable gate array (FPGA) in the coprocessor is not programmed to perform the function;

        the processor initiating an exception subroutine in response to the faulty commit;

        the exception subroutine fetching a configuration bit stream associated with the function for the application; and

        the exception subroutine performing a sequence of load and store instructions to program the field programmable gate array (FPGA) in accordance with the configuration bit stream to perform the function.

19.   (Cancelled)

20.   (Previously Presented) A system comprising:

a program memory;

a plurality of processors in communication with the program memory, each processor sharing the program memory among other ones of the plurality of processors;

a plurality of coprocessors separate from the plurality of processors, each coprocessor being coupled to a corresponding processor and including a field programmable gate array (FPGA); and

a shared resource manager operable to program each field programmable gate array associated with the plurality of coprocessors,

wherein a first processor of the plurality of processors is operable to execute an application and send an instruction from the program memory to each of the plurality of coprocessors to perform a function for the application,

if none of the field programmable gate array (FPGAs) associated with the plurality of coprocessors are programmed to perform the function for the application, then the shared resource manager is operable to dynamically program any one of the field programmable gate arrays (FPGAs) to perform the function for the application, in which the field programmable gate array (FPGA) selected to be dynamically programmed is selected in accordance with a least recently used algorithm, the least recently used algorithm specifying a function that can be disabled to free up logic resources within the field programmable gate array (FPGA) selected to be dynamically programmed.